EXHIBIT 6

Security & Identity

How to prevent account takeovers with new certificate-based access

September 17, 2024

Ratan Nalumasu
Director, Engineering, Google Cloud

Christopher Altman
Sr. Product Manager, Google Cloud

Stolen credentials are one of the top attack vectors used by attackers to gain unauthorized access to user accounts and steal information. At Google, we're continually evolving security capabilities and practices to make our cloud the most trusted cloud. To help protect your organization from stolen credentials, cookie theft, and accidental credential loss, we're excited to announce the general availability of certificate-based access in our Identity and Access Management portfolio.

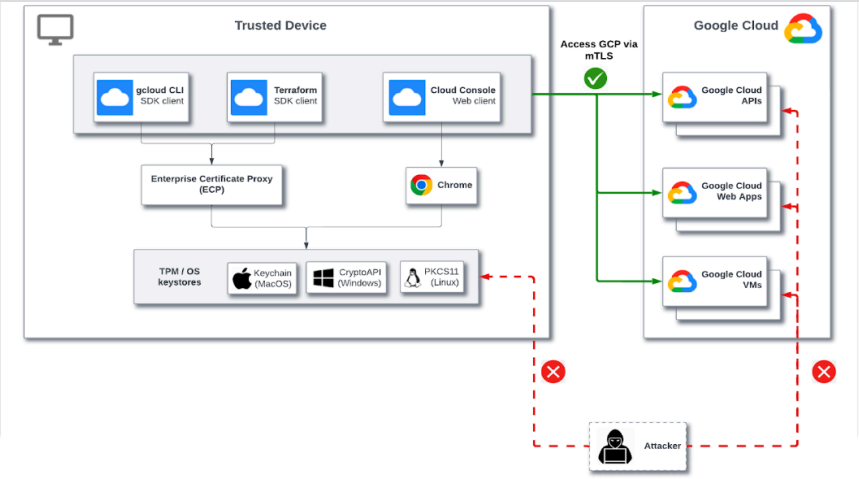
Certificate-based access (CBA) uses mutual TLS (mTLS) to ensure that user credentials are bound to a device certificate before authorizing access to cloud resources. CBA provides strong protection requiring X.509 certificates as device identifiers, and verifies devices with user context for every access request to cloud resources. Even if an attacker compromises a user's credentials, account access will remain blocked as they do not have the corresponding certificate. This renders the stolen credentials useless.

Mitigating account takeovers with certificate-based access

Certificate-based access can help you build a multi-layered defense against account takeovers, bolster data security, and maintain user trust. At Google, we have used this unique Zero Trust approach for many years as a strong defense to protect our technical infrastructure and employees. Now with CBA, we are extending this same level of security to our Google Cloud customers. Here are the key attributes of this approach:

- **Certificate-based access control**: Granular access policies with X.509 device certificates ensures only legitimate users with the correct certificate can access cloud resources.
- **Protections beyond initial login**: In contrast to using mTLS only for authentication, CBA also evaluates every authorization request to help safeguard resource access.
- **Strong key protection:** CBA's use of mTLS leverages secure cryptographic storage such as TPMs and OS keystores. Tooling such as Enterprise Certificate Proxy (ECP) are offered to users that empower them to safeguard private keys helping to ensure keys remain inaccessible to attackers without physical device access.

CBA allows you to enforce certificate-based authentication for all of Google Cloud, including specific resources (VM, Console, API), individual groups or organizations, across multiple end user vectors from browser to gcloud and terraform command-line-interfaces. It is integrated with many services in Google Cloud for effective access enforcement to cloud resources. For example, Context-Aware Access for Google APIs and the Cloud Console, Identity Aware Proxy (IAP) for web apps and workloads, and VPC Service Controls (VPC-SC) for network enforcement.



Certificate-based access (CBA) architecture overview.

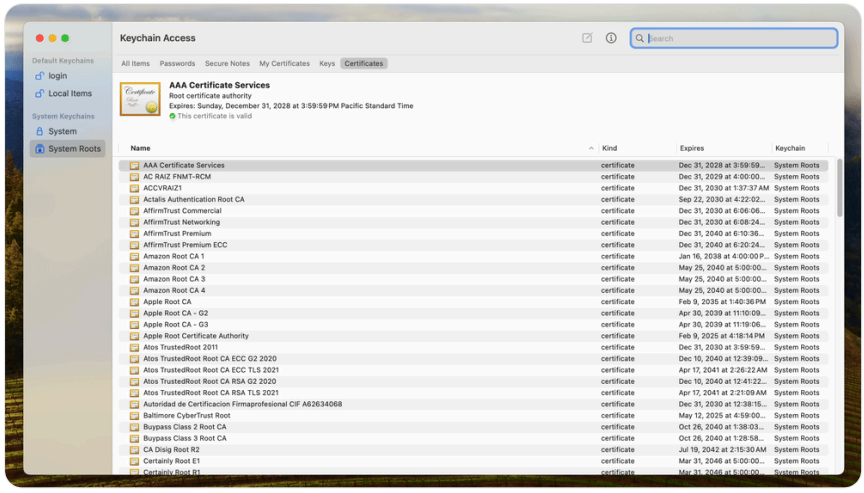
How to implement Certificate-based access

Certificate-based access operates on two core pillars:

- Access Levels: These act as the foundation for defining granular access control policies within Google Cloud. Access levels include a set of users and the access requirements necessary; an access level can require Certificates, but users can also define access levels using a wide variety of alternative signals.
- Certificate Identification: CBA relies on X.509 certificates as robust device identifiers. Google Cloud offers its own Certificate Authority Service, but most external Certificate Authority service providers are compatible as well.

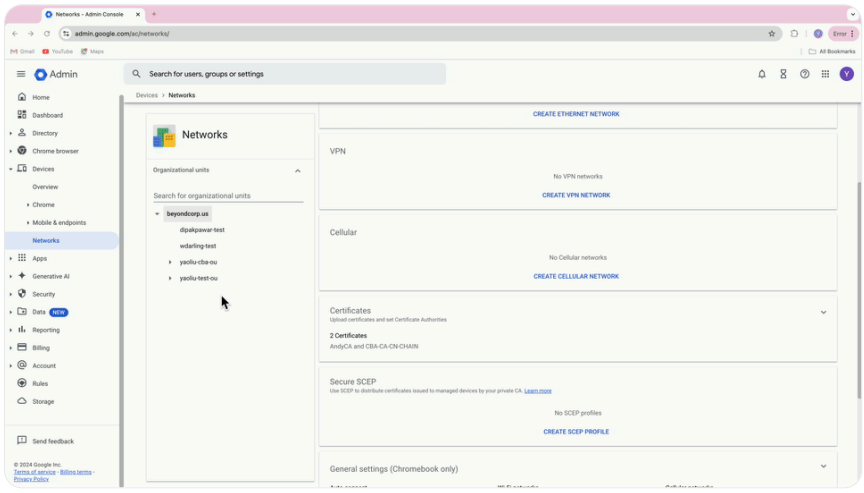
Step 1: Certificate provisioning
Choose the certificate options that work for you. We support customers' Certificate Authority providers (CA) and existing certificates on the devices. You can use Endpoint Verification certificates or Google's Certificate Authority Services for a fully-managed private CA.

We recommend the certificates to be stored securely in OS keystores or TPM and never expose the associated private keys to applications. CBA supports offloading private key operations to these secure storage via ECP and browser for applications built using Google Cloud SDK or web.



Step 2: Certificate registration

Administrators can configure an organization policy via the Google Admin console and instruct their corporate managed Chrome devices to register the chosen certificates to be accepted by CBA policy enforcement and use them in mTLS handshakes. The certificate registration is performed by the Endpoint Verification Chrome extension.

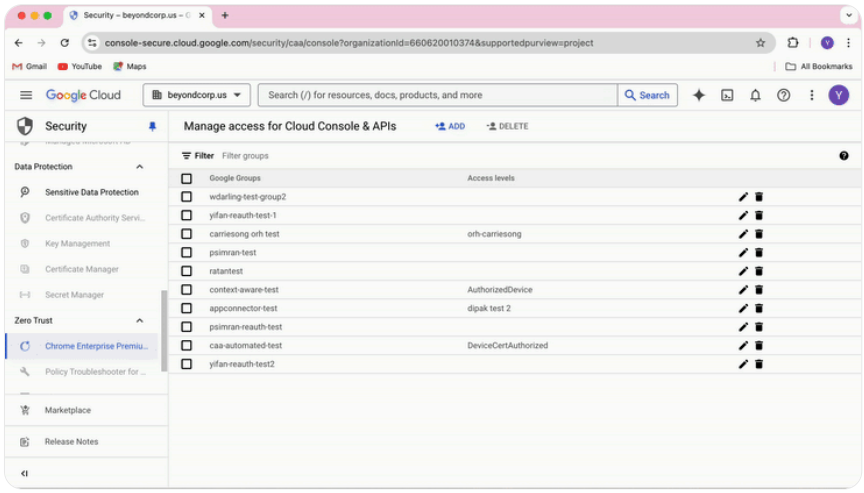


Certificate Registration using Endpoint Verification.

Step 3: Policy definition and enforcement

Last, admins will write CBA policies and enforce mTLS for accessing Google Cloud resources. The policy is implemented depending on the types of resources they want to protect. These policies apply to Google Cloud Console, gcloud CLI, Terraform, and third-party applications.

- Protecting access to Cloud Console and Google Cloud APIs
- Protecting access to Web Apps
- Protecting access to VMs



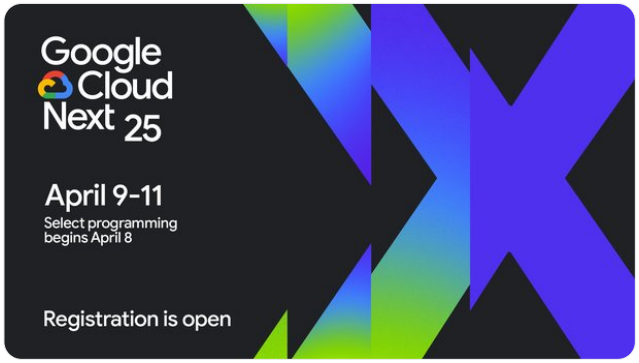
Create CBA Policy Enforcing mTLS

What's next

means of identifying your users. To learn more about implementing CBA as part of your overall security strategy for Google Cloud, please refer to our documentation.

Posted in Security & Identity

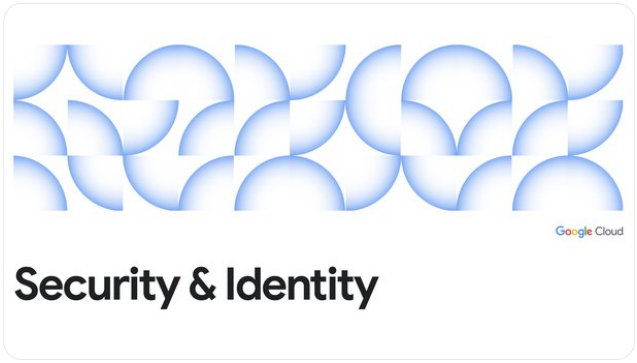
Related articles



Security & Identity

Get ready for a unique, immersive security experience at Next '25

By Robert Sadowski • 5-minute read



Security & Identity

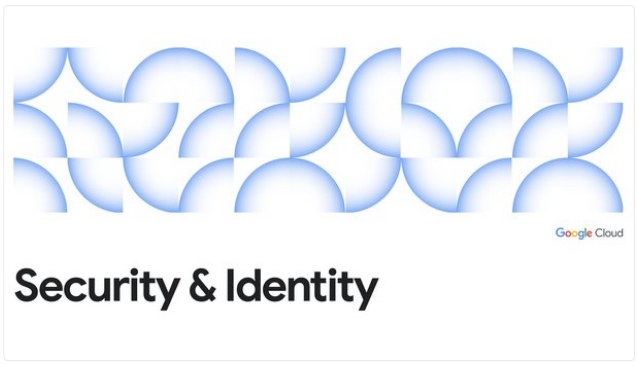
Introducing Google Cloud Abuse Event Logging to enable automated incident remediation

By Alex Dininger • 4-minute read

Security & Identity

Cloud CISO Perspectives: From gen AI to threat intelligence: 2024 in review

By Phil Venables • 5-minute read



Security & Identity

How Google Cloud can help customers achieve compliance with NIS2

By Tara Brady • 9-minute read